Amendments In the Claims

Please amend Claims 1-7, 11, 12, 14, 16, 19, 21, 24-29, 33-35, 39-41, 44-47 and 51-53 as follows:

1.　　(**Currently Amended**)　A method of arranging **a plurality of** objects**,** **said method** comprising:

setting a class hierarchy, wherein

the class hierarchy comprises an upper level class and a lower level class[[,]] **;** ~~and~~

~~the objects are members of at least one of the upper level class and the lower level class;~~

assigning a first attribute to the upper level class, wherein

the first attribute describes ~~the~~ **each** object[[s]] **of the plurality of objects**;

inheriting of the first attribute by the lower level class, wherein

the first attribute is within a first domain with regard to the upper level class,

the first attribute is within a second domain with regard to the lower level class,

a second domain value set of the second domain is smaller than a first domain value set of the first domain, and

the first attribute is restricted to the second domain value set upon the inheriting;

assigning a second attribute to the lower level class, wherein

the second attribute describes ~~an~~ **each** object **of the plurality of objects** associated with the lower level class;

associating an object of the plurality of objects with a chosen class, wherein

the chosen class is one of the upper level class and the lower level class, and

the chosen class is chosen such that, for the object, every attribute assigned to the chosen class has a non-null value used to describe the object; and

~~choosing a class with which to associate an object, wherein the class is chosen such that every attribute associated with the class has a non-null value used to describe the object; and~~

said method is performed by a processor configured to perform said method.

2.　**(Currently Amended)** The method of ~~arranging objects of~~ claim 1, further comprising:

superseding said first attribute of said upper level class by assigning a third attribute to the lower level class, wherein the third attribute describes an object that is associated with the lower level class.

3.　**(Currently Amended)** The method of ~~arranging objects of~~ claim 1, wherein the first attribute comprises a distinctive domain value set.

4.　**(Currently Amended)** The method of ~~arranging objects of~~ claim 1, wherein the class hierarchy further comprises a third class below the lower level class in the class hierarchy, and further comprising:

inheriting of the first attribute and the second attribute by the third class.

5.　**(Currently Amended)** The method of ~~arranging objects of~~ claim 1, further comprising:

expanding the class hierarchy horizontally by adding a fourth class to the lower level class; and

inheriting of the first attribute by the fourth class.

6.　**(Currently Amended)** A hierarchical class architecture of objects stored in a memory**, said hierarchical class architecture** comprising:

an upper level class;

- 3 -

a lower level class, wherein the upper and lower level classes are stored in the

      memory;

a first domain value set of a first domain of the upper level class;

a second domain value set of a second domain;

a first attribute, wherein

      the first attribute is assigned to the upper level class,

      the first attribute is within the first domain,

      the first attribute is within the second domain,

      ~~the objects are members of at least one of the upper level class and the~~

            ~~lower level class,~~

      the first attribute describes **each** ~~the~~ object[[s]],

      the lower level class is configured to inherit the first attribute,

      the second domain value set is smaller than the first domain value set, and

      the first attribute is restricted to the second domain value set upon the first

            attribute being inherited by the lower level class; and

a second attribute, wherein

      the second attribute is assigned to the lower level class,

      the second attribute is within the second domain, and

      the second attribute describes **each** ~~an~~ object associated with the lower

            level class, and

      each object in the hierarchical class architecture of objects is associated

            with a chosen class within the class hierarchy, wherein

            <u>**the chosen class is one of the upper level class and the lower**</u>

                  <u>**level class, and**</u>

            the class is chosen such that<u>**, for each object associated with the**</u>

                  <u>**chosen class,**</u> every attribute <u>**assigned to**</u> ~~associated with~~

                  the <u>**chosen**</u> class has a non-null value used to describe the

                  object.

7.   **(Currently Amended)** The hierarchical class architecture of claim 6, further comprising:

an additional attribute, wherein

the additional attribute is assigned to the lower level class, and

the **additional** attribute describes **each** ~~an~~ object **associated with** ~~in~~ the lower level class.

8.   (Previously Presented) The hierarchical class architecture of claim 6, wherein the first attribute comprises a distinctive domain value set.

9.   (Previously Presented) The hierarchical class architecture of claim 6, further comprising:

a third class, wherein

the third class is below the lower level class in the hierarchical class architecture, and

the third class is configured to inherit the first attribute and the second attribute.

10.   (Previously Presented) The hierarchical class architecture of claim 6, wherein

the lower level class is configured to be expanded horizontally by virtue of being configured to provide for addition of a fourth class, and

the fourth class is configured to inherit the first attribute.

11.   **(Currently Amended)** A computer system comprising:

a processor;

a computer readable medium coupled to the processor; and

computer code, encoded in the computer readable medium, configured to cause the processor to:

set a class hierarchy, wherein

the class hierarchy comprises an upper level class and a lower level class**;** ~~, and~~

- 5 -

the objects are members of at least one of the upper level class
and the lower level class;

assign a first attribute to the upper level class, wherein

the first attribute describes <u>each</u> ~~the~~ object[[s]] <u>of a plurality of
objects</u>;

provide inheritance of the first attribute by the lower level class, wherein

the first attribute is within a first domain with regard to the upper
level class,

the first attribute is within a second domain with regard to the
lower level class,

a second domain value set of the second domain is smaller than a
first domain value set of the first domain, and

the first attribute is restricted to the second domain value set upon
the inheritance of the attribute by the lower level class;

assign a second attribute to the lower level class, wherein

the second attribute describes <u>each</u> ~~an~~ object <u>of the plurality of
objects that is</u> associated with the lower level class; and

<u>associate an object of the plurality of objects with a chosen class</u>
~~choose a class with which to associate an object~~, wherein

<u>the chosen class is one of the upper level class and the lower
level class, and</u>

the class is chosen such that<u>, for the object,</u> every attribute
<u>assigned to</u> ~~associated with~~ the <u>chosen</u> class has a non-
null value used to describe the object.

12.   (Currently Amended) The computer system of claim 11, wherein the
computer code is further configured to cause the processor to:

assign a third attribute to the lower level class, the third attribute <u>describes each</u>
~~describing an~~ object <u>of the plurality of objects that is associated with</u>
~~that is a member of~~ the lower level class.

- 6 -

13.   (Previously Presented) The computer system of claim 11, wherein the first attribute comprises a distinctive domain value set.

14.   **(Currently Amended)** The computer system of claim 11, wherein ~~the class hierarchy further comprises a third class below the lower level class in the class hierarchy, and~~ the computer code is further configured to cause the processor to:

provide inheritance of the first attribute and the second attribute by <u>a</u> ~~the~~ third class<u>, wherein</u>

<u>the third class is below the lower level class in the class hierarchy</u>.

15.   (Previously Presented) The computer system of claim 11, wherein the computer code is further configured to cause the processor to:

expand the class hierarchy horizontally by adding a fourth class to the lower level class; and

provide inheritance of the first attribute by the fourth class.

16.   **(Currently Amended)** An apparatus for arranging objects comprising:

means for setting a class hierarchy, wherein

the class hierarchy comprises an upper level class and a lower level class[[,]] ~~and~~

~~the objects are members of at least one of the upper level class and the lower level class~~;

means for assigning a first attribute to the upper level class, wherein the first attribute describes <u>each</u> ~~the~~ object[[s]];

means for inheriting of the first attribute by the lower level class, wherein

the first attribute is within a first domain with regard to the upper level class,

the first attribute is within a second domain with regard to the lower level class,

a second domain value set of the second domain is smaller than a first domain value set of the first domain, and

the first attribute is restricted to the second domain value set by the means

for inheriting;

means for assigning a second attribute to the lower level class, wherein

the second attribute describes **each** object[[s]] associated with the lower

level class; and

means for **associating an object with a chosen class** ~~choosing a class with~~

~~which to associate an object~~, wherein

the chosen class is one of the upper level class and the lower level

class, and

the class is chosen such that**, for the object,** every attribute **assigned to**

~~associated with~~ the **chosen** class has a non-null value used to

describe the object.

17.   (Previously Presented) The apparatus of claim 16, further comprising:

means for superseding said first attribute of said upper level class comprising

means for assigning a third attribute to the lower level class, wherein

the third attribute describes an object that is associated with the lower

level class.

18.   (Previously Presented) The apparatus of claim 16, wherein the first

attribute comprises a distinctive domain value set.

19.   (**Currently Amended**) The apparatus of claim 16~~, wherein the class~~

~~hierarchy further comprises a third class below the lower level class in the class~~

~~hierarchy, and~~ further comprising:

means for inheriting of the first attribute and the second attribute by **a** ~~the~~ third

class**, wherein**

the third class is below the lower level class in the class hierarchy.

20. (Previously Presented) The apparatus of claim 16, further comprising:

means for expanding the class hierarchy horizontally by adding a fourth class to the lower level class; and

means for inheriting of the first attribute by the fourth class.

21. **(Currently Amended)** A computer program product, encoded in computer readable media, comprising:

a first set of instructions, executable on a computer system, configured to set a class hierarchy, wherein

the class hierarchy comprises an upper level class and a lower level class[[,]] ~~and~~

~~the objects are members of at least one of the upper level class and the lower level class~~;

a second set of instructions, executable on the computer system, configured to assign a first attribute to the upper level class, wherein the first attribute describes **each** ~~the~~ object[[s]] **of a plurality of objects**;

a third set of instructions, executable on the computer system, configured to provide inheritance of the first attribute by the lower level class, wherein

the first attribute is within a first domain with regard to the upper level class,

the first attribute is within a second domain with regard to the lower level class,

a second domain value set of the second domain is smaller than a first domain value set of the first domain, and

the first attribute is restricted to the second domain value set by the third set of instructions;

a fourth set of instructions, executable on the computer system, configured to assign a second attribute to the lower level class, wherein

the second attribute describes **each** ~~an~~ object **of the plurality of objects that is** associated with the lower level class; and

a fifth set of instructions, executable on the computer system, configured to

    associate an object of the plurality of objects with a chosen class

    ~~choose a class with which to associate an object~~, wherein

        the chosen class is one of the upper level class and the lower

        level class, and

    the class is chosen such that**, for the object,** every attribute

        associated with the **chosen** class has a non-null value used

        to describe the object.

22.   (Previously Presented) The computer program product of claim 21, further comprising:

    a sixth set of instructions, executable on the computer system, configured to

        supersede said first attribute of said upper level class by virtue of being

        configured to assign a third attribute to the lower level class, the third

        attribute describing an object that is associated with the lower level class.

23.   (Previously Presented) The computer program product of claim 21, wherein the first attribute comprises a distinctive domain value set.

24.   (**Currently Amended**) The computer program product of claim 21~~,~~ ~~wherein the class hierarchy further comprises a third class below the lower level class in the class hierarchy, and~~ further comprising:

    a sixth set of instructions, executable on the computer system, configured to

        provide inheritance of the first attribute and the second attribute by **a** ~~the~~

        third class**, wherein**

        the third class is below the lower level class in the class hierarchy.

25.   (**Currently Amended**) The computer program product of claim 21, further comprising:

    a ~~an~~ sixth set of instructions, executable on the computer system, configured to

        expand the class hierarchy horizontally by adding a fourth class to the

        lower level class; and

a seventh set of instructions, executable on the computer system, configured to

provide inheritance of the first attribute by the fourth class.

26. **(Currently Amended)** The method of ~~arranging objects of~~ claim 1, further comprising:

associating the upper level class with the first domain value set, and

associating the lower level class with the second domain value set.

27. **(Currently Amended)** The method of ~~arranging objects of~~ claim 26, wherein

a third ~~another~~ attribute is within the second domain.

28. **(Currently Amended)** The method of ~~arranging objects of~~ claim 27, wherein

the **third** ~~another~~ attribute is an overriding attribute.

29. **(Currently Amended)** The method of ~~arranging objects of~~ claim 27, further comprising:

superceding the **first** attribute with the **third** ~~another~~ attribute, wherein

the superceding is performed if the second domain is different from the

first domain.

30-32. Cancelled.

33. **(Currently Amended)** The hierarchical class architecture ~~of objects~~ of claim 6, further comprising:

a third ~~another~~ attribute, wherein the **third** ~~another~~ attribute is ~~another attribute~~

within the second domain.

34. **(Currently Amended)** The hierarchical class architecture ~~of objects~~ of claim 33, wherein

the **third** ~~another~~ attribute is an overriding attribute.

35. **(Currently Amended)** The hierarchical class architecture ~~of objects~~ of claim 33, wherein

the **third** ~~another~~ attribute is configured to supercede the **first** attribute, and

the **third** ~~another~~ attribute **is configured to** supercede[[s]] the **first** attribute if the second domain is different from the first domain.

36-37. Cancelled.

38. (Previously Presented) The computer system of claim 11, wherein the computer code is further configured to cause the processor to:

associate the upper level class with the first domain value set, and

associate the lower level class with the second domain value set.

39. **(Currently Amended)** The computer system of claim 38, wherein **a third** ~~another~~ attribute is within the second domain.

40. **(Currently Amended)** The computer system of claim 39, wherein the **third** ~~another~~ attribute is an overriding attribute.

41. **(Currently Amended)** The computer system of claim 39, wherein the computer code is further configured to cause the processor to:

supercede the **first** attribute with the **third** ~~another~~ attribute, if the second domain is different from the first domain.

42-43. Cancelled.

44. **(Currently Amended)** The apparatus of claim 16 **further comprising,** ~~wherein the computer code is further configured to cause the processor to~~:

means for associating ~~associate~~ the upper level class with the first domain value set**;** [[,]] and

means for associating ~~associate~~ the lower level class with the second domain value set.

45.   (**Currently Amended**) The apparatus of claim 44, wherein
a third ~~another~~ attribute is within the second domain.

46.   (**Currently Amended**) The apparatus of claim 45, wherein
the **third** ~~another~~ attribute is an overriding attribute.

47.   (**Currently Amended**) The apparatus of claim 45, further comprising:
means for superceding the **first** attribute with the **third** ~~another~~ attribute, wherein
        the superceding is performed if the second domain is different from the
                first domain.

48-49. Cancelled.

50.   (Previously Presented) The computer program product of claim 21, further
comprising:
        a sixth set of instructions, executable on the computer system, configured to
                associate the upper level class with the first domain value set; and
        a seventh set of instructions, executable on the computer system, configured to
                associate the lower level class with the second domain value set.

51.   (**Currently Amended**) The computer program product of claim 50,
wherein
        a third ~~another~~ attribute is within the second domain.

52.   (**Currently Amended**) The computer program product of claim 51,
wherein
        the **third** ~~another~~ attribute is an overriding attribute.

53.   (**Currently Amended**) The computer program product of claim 51,
further comprising:

an eighth ~~a twelfth~~ set of instructions, executable on the computer system,

configured to supercede the **first** attribute with the **third** ~~another~~

attribute, if the second domain is different from the first domain.

54-55. Cancelled.